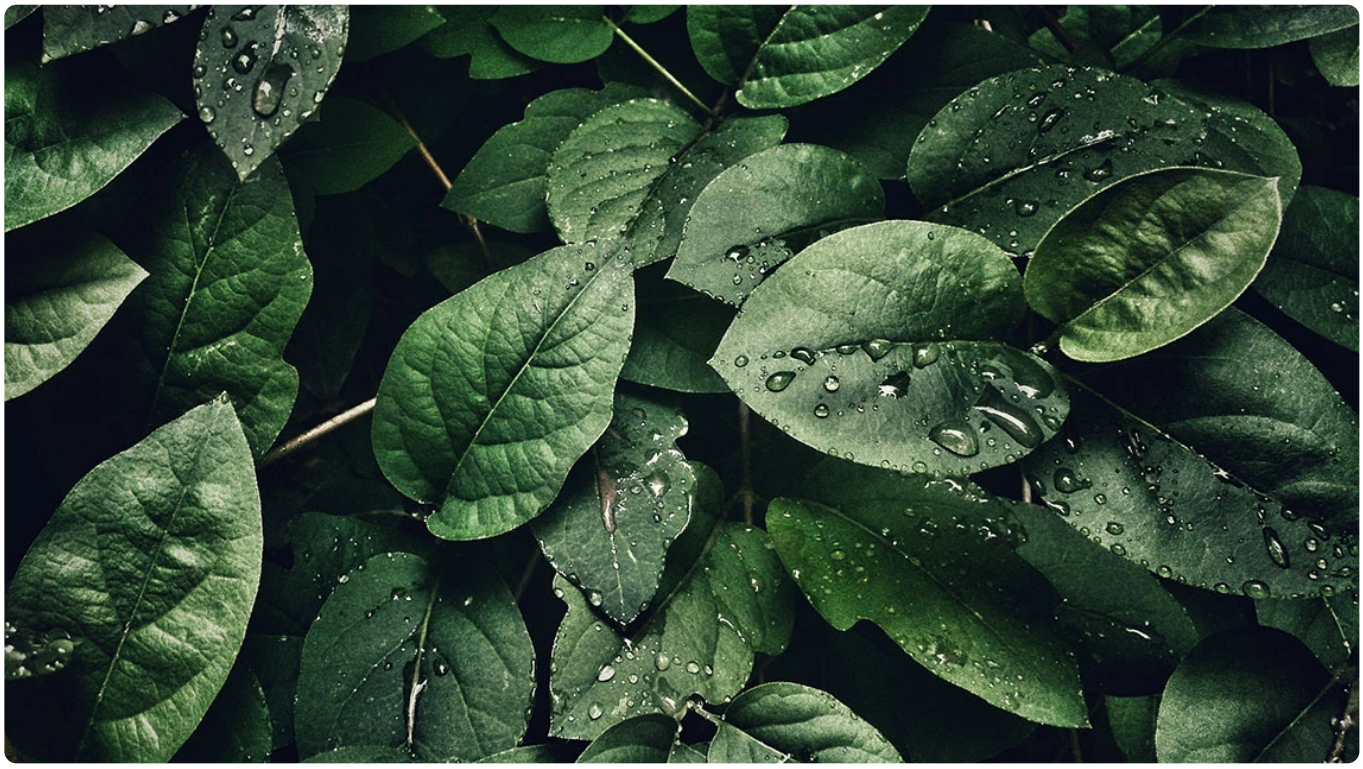
下载APP

09｜音频编解码器是如何工作的？

2021-12-10 冯建元

《搞定音频技术》 课程介绍 >



讲述：冯建元

时长 18:04　大小 16.56M

你好，我是建元。今天我们来聊聊什么是音频编 / 解码器，以及它背后的原理。

在我们讲音频编 / 解码器之前，你不妨先设想一下：如果没有音频编 / 解码器，我们直接给对方发送原始数据，会发生什么事情呢？

我们假设一个在线会议有 10 个人，每个人要发给另外 9 个人的音频信号是 48kHz 采样率的单通道音频。每个采样点我们用 16 位的浮点来表示。那么上行通道中我们需要每秒发送 48000 乘以 16 bit 的音频信号，也就是大约 768kbps；而接收的下行通路的信号是除了你之外的另外 9 个人的信号，也就是 6912kbps。

所以上行加下行要想在没有音频编 / 解码的情况下，完成一个 10 人的在线会议，你需要消耗 7.5Mb 的带宽在音频信号上，来保证流畅的音频通话，注意这里我们还没有计算发送的

包头以及抗弱网所需要的额外带宽。

这么大的带宽消耗在带来巨大成本的同时，也会在网络条件比较差或者通话人数比较多的时候让实时音频出现卡顿甚至不可用的问题。所以音频编 / 解码器的任务就是在减少带宽码率消耗的同时，尽量保持音频的音质不会受到损伤。

那么具体是如何实现的呢？这节课我们就来说道说道。清楚了这些，对于你接下来实践音频编解码，以及选择合适的编解码器都是大有帮助的。

编 / 解码器的发展史

这里我们从发展的视角看下，音频编 / 解码器都有哪些类型，就以它的作用来区分吧。

音频编 / 解码器包括编码和解码两个部分。编码的过程就是对音频进行压缩，而**压缩的过程是为了保留音频的主要甚至全部信息**。解码就是对压缩后的音频信号进行解码，从而恢复原始的音频信号。

压缩按照是否可以完美还原，可以分为无损压缩和有损压缩。无损压缩，例如 APE、FLAC 等格式可以让音频中所有的细节都得到还原，而有损压缩可提供更低的码率，是我们在实时音频中使用更多的压缩类型。

一段音频包含的信息其实可以有很多。比如，里面可能有语音、乐器、噪声等多种信号，而其中的语音部分，是我们平时实时音频互动中最重要的部分。

最早的一批实时音频编 / 解码器，如**基于 ITU 标准的 G.71 等就是针对如何保持语音部分而设计的。我们把这一类音频编 / 解码器叫做语音编 / 解码器**。

后来为了传输更多的信息，比如包括音乐甚至“噪声”等全部音频信号的编 / 解码器，例如基于 MPEG 标准的 MP3 和 AAC 也陆续出现。

随后，**主要用来编码语音信号的语音编 / 解码器，逐渐向基于时域的线性预测框架的方向演化。**这种编 / 解码器参考了声道的发音特性，将语音信号分解为主要的线性预测系数和次要的残差信号。而线性预测系数编码所需的比特率非常少，却能高效地构建出语音信号的“骨骼”；残差信号则像是“血肉”，能够补充出语音信号的细节。这种设计大幅提升

了语音信号的压缩效率。**但是这种基于时域的线性预测框架在有限的复杂度下无法很好地编码音乐信号。**

因此，**针对音乐信号进行编码的音乐编 / 解码器走上了另一条演化的道路**。因为相比时域信号，频域信号的信息更多集中在少部分频点上，更利于编码器对其进行分析和压缩。所以**音乐编 / 解码器基本都会选择在频域上对信号进行频域编码**。比如图 1 中基本上语音的频谱从下到上是连续的（频谱上呈现为一团红色或者一些横跨不同频率的曲线），音乐信号则在部分频段上有一些持续性的能量（频谱上呈现为一条一条的直线）。

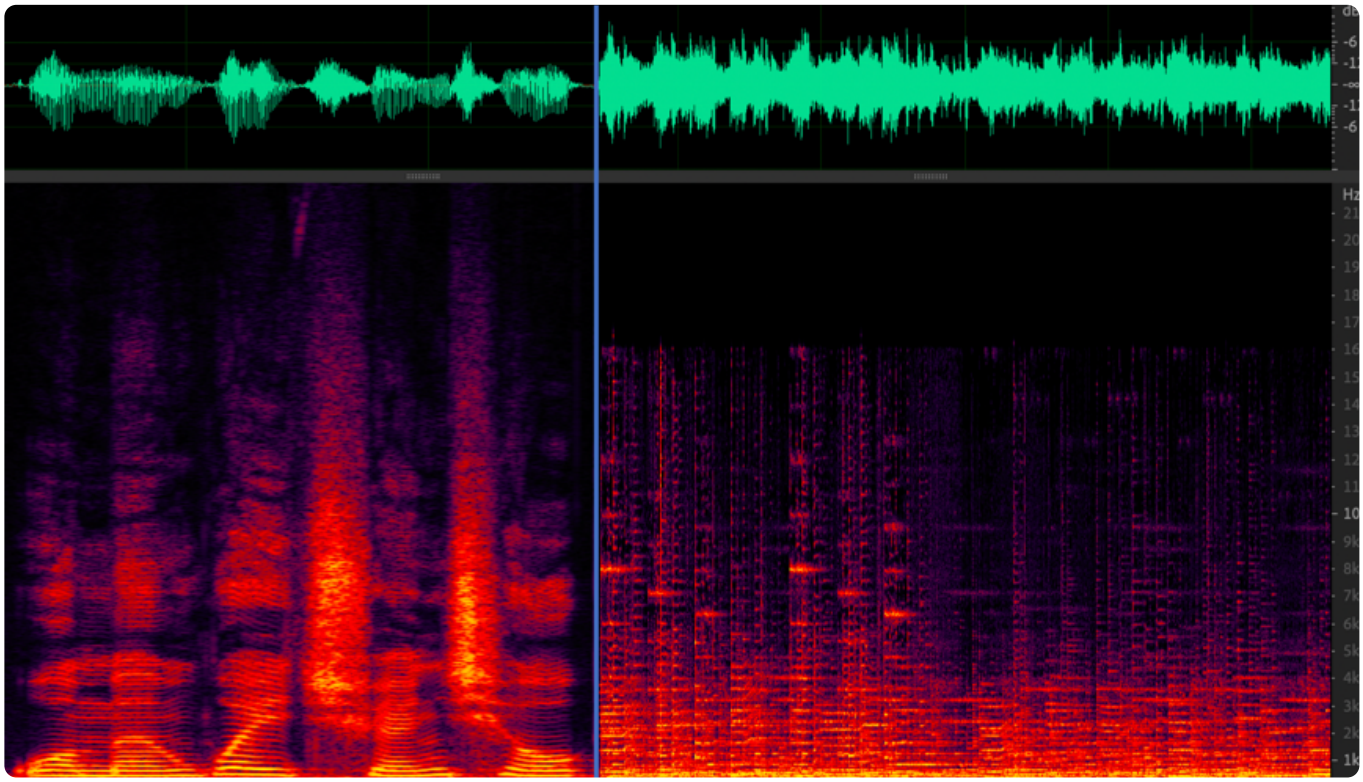


图1 人声（左）和音乐（右）波形和频谱图

后来，随着技术日趋成熟，**两种编 / 解码架构又再次走到了一起，即语音音乐混合编码器。**WebRTC 中默认使用的编 / 解码器 OPUS 就是这类编 / 解码器。**这类编 / 解码器的特点是融合了两种编码框架，并针对信号类型自动切换合适的编码框架。**一些国内外知名的产品都会用到 OPUS，比如 Discord。

会议等主要以人声为主的场景可以使用语音编解码器省流量，而音乐直播等场景则需要更多的流量来保证音质，或者两者都有的情况下可以选用 OPUS 这样的来自动调节，具体如何选择编 / 解码器会在下一讲中详细介绍。这里会先介绍一些通用的方法比如音频数据裁剪和量化，再举例介绍一下语音和音乐编解码器的基本原理。

编 / 解码的算法细节比较多，这里我们主要介绍一些常见的方法，看看我们是如何一步步的把音频的码率降下来，而又不损伤音质的。

音频数据裁剪和量化

还记得我们一开始举的那个例子么，即 48kHz、每个采样点用 16bit 的浮点来表示的音频。实际上，音频压缩的第一步就是从这些音频的参数出发来缩减采样率和采样精度非线性量化。不懂？没关系，继续往下看。

先了解点基础知识，根据编码音频的带宽，我们可以把音频分为窄带、宽带、超宽带和全通带。下表中我列出了带宽和采样频率的关系。

带宽和采样频率的关系				
带宽	窄带	宽带	超宽带	全通带
采样频率	8kHz	16kHz	32kHz	48kHz

极客时间

举个例子帮助你理解，你可以关注一下打电话这个场景，打电话的时候我们能听得见对方在说什么，但是往往会觉得声音有点闷。这是因为打电话时语音信号的采样率实际上只有 8kHz，也就是窄带信号，根据采样定律这里的有效频谱是 0～4kHz。所以说，在低采样率的情况下，语音信号中的语音信息被有效保留了，而更高频的音色信息没有被保留。总结来说，就是采样率降低了，每秒发出的数据量也就降低了，这就是通过缩减采样率来压缩了音频。

那么采样精度非线性量化又是什么意思呢？还是回到最开始的例子，我们每个采样点用 16bit 的浮点来表示，而如果换做非线性量化来表示可能只需要 8bit 就可以，从听感上来说却不会有太大的损失。这是为什么呢？

比如，**G.711 中的 A-LAW 就是采用非线性量化的方式对每个采样点进行压缩。**而在这里用 8bit 量化的采样点表示就可以比 16bit 的少一半的数据量。在图 1 的语音时域信号里我们可以发现，大部分信号的幅度都在比较低的范围内，只有少部分的幅度值会比较大。如果我们给予更多的精度用于描述低幅度的信号，那么压缩后，听感上的损失就会比较小。比如下面 A-LAW 公式中就是通过固定值 $1/A$ 作为区分幅度大小的的界限，用非线性的对数 ln 来划分表达的幅度位置，对高、低幅度分别用不同的量化精度对采样点进行量化。

$$f(x) = \begin{cases} A(x)/(1+lnA), & 0 <= x <= 1/A \\ (1+lnAx)/(1+lnA), & 1/A <= x <= 1 \end{cases} \text{A-LAW 解码}$$

语音编 / 解码器的基本原理

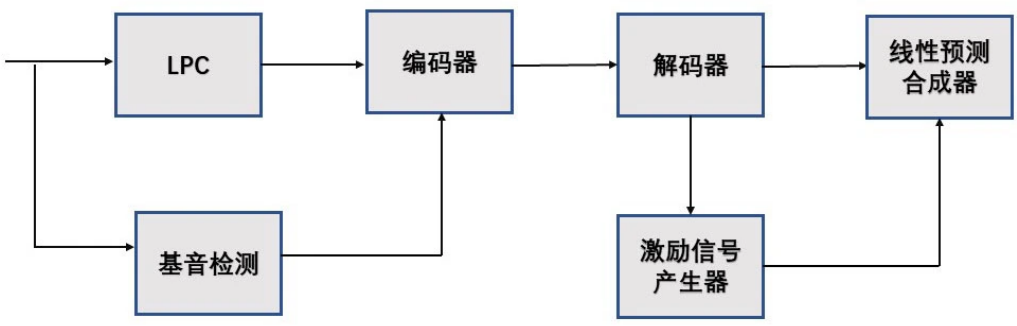
上述缩减采样率和采样精度非线性量化都是数据层面的压缩，接下来我们再看看如何从语音合成建模的角度对语音进行编 / 解码。这里我们常采用线性预测编码（LPC）。

这里你可以先回想一下我们之前讲的人发声的基本原理。人类发声时，声带振动得到的是浊音，而声带不震动并且通过气流吹过唇齿等部位产生的声音为清音。音调或者说音高就表现为基频 F0 的频率高低，其中这个基频是声带振动的频率。而音色则是由频谱的包络来决定的。频谱包络反映的是咽喉、嘴、鼻等声道的形状引发的共振信息，所以频谱包络也就是音效中常说的共振峰曲线。

再联想一下前面我们讲语音信号分析时，我们发啊、哦、额的时候，我们的口型是不是不一样的。因此，在语音的建模中，模拟声道建模的线性预测滤波器就可以派上用处了。线性滤波器的系数（LPC）反映的是各个声道器官的状态。而在 10～30ms 时间内器官移动的距离有限，所以我们可以认为声道器官的位置基本不变化，这样就可以利用线性滤波器来对语音进行编码。而线性滤波器的系数远比时域对应帧的采样点个数要少很多，比如我们一般用 16 阶的 LPC 就可以来表示一个 10ms、16kHz 采样的 160 点的帧。LPC 的公式如下所示，我们可以看到这是一个自回归的模型，即当前值是过去值的加权预测。

$$y(n) = a_1 y(n-1) + a_2 y(n-2) + a_3 y(n-3) + \ldots + e(n) \text{ LPC 模型}$$

基于 LPC 的编 / 解码器的结构如图 2 所示：



极客时间

图2 基于LPC的编/解码器

其中，基音检测是 LPC 类型的编 / 解码器必备的模块之一。**基音检测模块会提供两个信息：一个是信号的周期性，也就是区分是清音还是浊音；另一个是如果是浊音那么基频是多少 Hz。**如果判断为浊音，那么激励信号产生器会产生一个和基频相同周期的脉冲激励信号；如果是清音则产生一个白噪声，然后由线性预测合成器按照频谱包络来还原出原始音频。

LPC 模型是线性预测，有预测就会有预测误差。值得注意的是在 LPC 模型的公式里我们可以看到还有一项 $e(n)$，我们把这一项叫做噪声或者残差（excitation）。由于音频信号不可能是完全线性的，如果这个部分缺失，你依然可以听清楚一个语音发的是什么音，但声音会非常“生硬”，或者说听着像机器人。因此，我们实际使用 LPC 模型时还需要对残差部分来进行编码传输。也就是说在编码传输的时候我们需要传递 LPC 系数和残差这两个部分，才能在解码时将语音比较好的还原出来。

例如，在实际使用中，由 Skype 公司发明的基于 LPC 的 SILK 编 / 解码器使用长时预测分析（LPT）来估计残差信号，使用 Burgs 方法来计算 LPC 系数。LPC 系数被转换为线谱频率（LSF）向量，然后对 LSF 进行矢量量化（VQ，Vector Quantization），在解码的时候再结合残差信号把 LSF 转换为 LPC 系数。这样就实现了残差和 LPC 系数的编码。如果直接用 LPC 系数做量化，一两个系数的误差可能会导致某些频段有较大的失真。在这里，LSF 是对 LPC 模型的因式分解形式，在后续做量化的时候可以减少单一频段的失真。

对于类似 LPC 的参数编码器，可以注意到如果类似 LPC 的参数不进行进一步编码，每一帧仍需要传一组 LPC 参数或 LSF 参数。如果 1 秒有 100 帧，那么这个码率还是有点大的。例如 SILK 中就使用了多级向量码本的方式来解决这个问题。 我们通常把这个过程叫矢量量化 VQ。

VQ 是一种基于块编码规则的有损数据压缩方法。事实上，在音视频中的 JPEG 和 MPEG-4 等多媒体压缩格式里都有 VQ 这一步。它的基本思想是：将若干个标量数据组构成一个矢量，然后在矢量空间给以整体量化，从而压缩了数据。

比如**编码器中每一帧都有一组需要编码的参数向量，那这些参数就可以通过有限数量的固定向量来表达。**比如固定 1024 种参数的组合，这时如果有新的需要量化的参数向量则需要找到和它最接近的那个固定向量来表达。这个固定的 1024 个参数向量我们把它编成一个码本，且码本的标号 1 至 1024 分别代表这 1024 种参数组合。那么在编码的时候只需

要从码本中找到与这个最接近的参数向量的编号，比如 16，然后把 16 发给解码器。解码端有一个同样的码本。解码的时候只需要去找第 16 个编号代表的参数向量就可以实现解码。这样传输过程中只用到 16 这一个数，而参数向量一共有 1024 种可能，所以编码的时候使用 11bit 来编码就可以了。

VQ 所需的码本是通过大量的离线语料训练得到的。VQ 的训练方法有很多，常见的有 LBG-VQ，K-means 等，这里我们不再详细介绍。**其中 SILK 所用的是多级码本**。这里的多级码本是指第一阶段的输入是待量化的矢量，而之后每一级的输入是前一级的输出的量化误差。比如一个三级码本，编码的时候需要传 3 个编号，解码的时候则分别根据编号查三个码本，然后把三个码本的向量相加，从而得到完整的解码结果。

音乐编 / 解码器的基本原理

上面这些主要是语音编码器的基本原理。我们可以看到**语音编码器主要是对语音的发声来建模编 / 解码**。而音乐编 / 解码器因为要编码频带更丰富的音乐信号，所以更多的是**从听得清晰的角度利用心理听觉来进行编码**。也就是说我们人耳更敏感的频带需要多耗费一些码率来编码，不敏感的则少耗费一些码率。

这里我们以 CELT 编 / 解码器作为例子。CELT 编码全称是 Constrained Energy Lapped Transform，主要使用 MDCT 编码。离散余弦变换（DCT）在音、视频编码中都是常用的降维方法，对其内部原理有兴趣的同学可以自行了解一下。MDCT 其实就是时域重叠的 DCT 变换，主要是为了消除 DCT 带来的块效应。MDCT 的正、逆变换可以参照图 3。

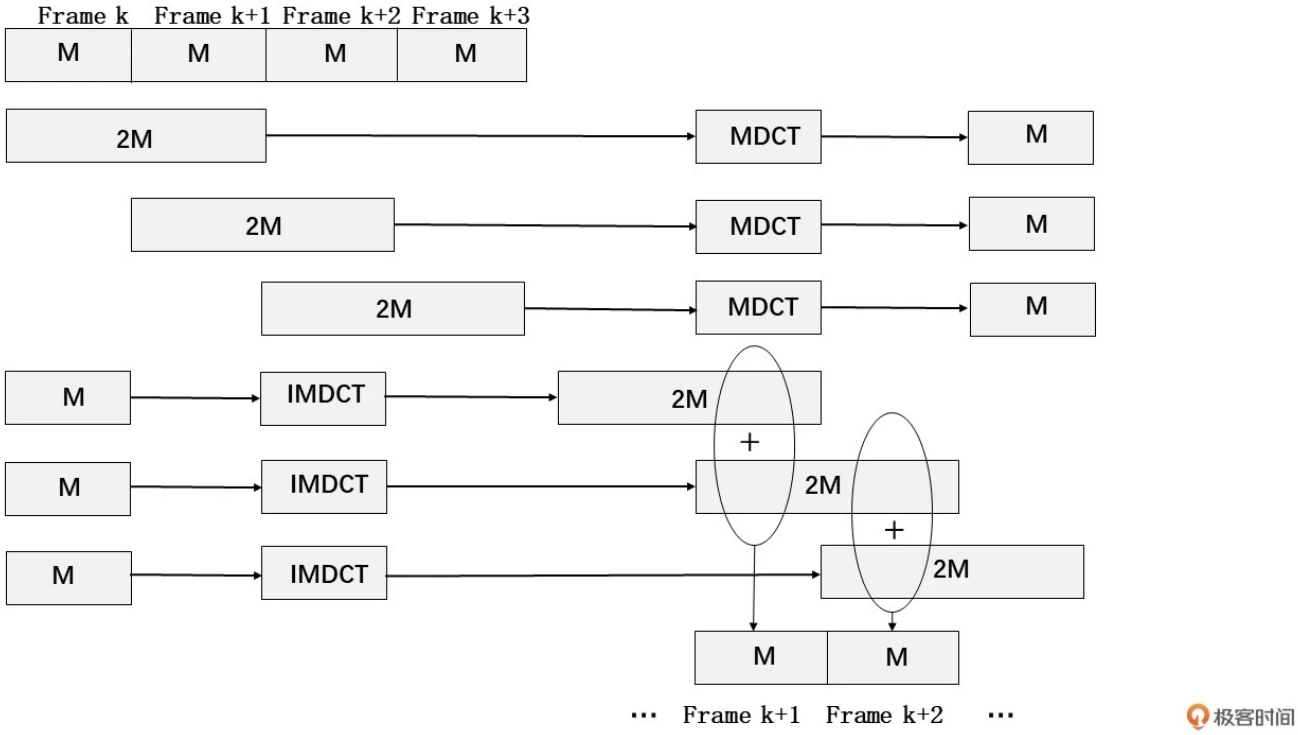


图3 MDCT和IMDCT

MDCT 编码主要过程是分帧，然后一次有重叠的取几帧（比如图 3 是一次取 2 帧）做 MDCT 变换得到一个编码信息。解码时，每个编码信息通过 IMDCT 逆变换还原编码的那几帧的信号，然后再通过滑动叠加的方式得到还原的音频信号。

之前的课程里我们讲过人对不同的频带的感知是不同的，比如，人对低频较高频的频率变化较为敏感。回想一下我们之前讲过的人耳敏感频响曲线，Bark 谱和 Mel 谱。在音乐这种频带分布比较随机的信号，就需要对不同的频段加以划分，有的频段人的听感比较敏感就需要更细致的编码，相反则可以进行加大程度的压缩。

比如 CELT 中的 MDCT 变换频谱就是基于 Bark 频带的，并且选了约 21 个频带进行编码。每个频带都进行分析、量化数据，并通过预测压缩，可以直接将差异传到预测值。Bark 谱的频带分布可以参照图 4。

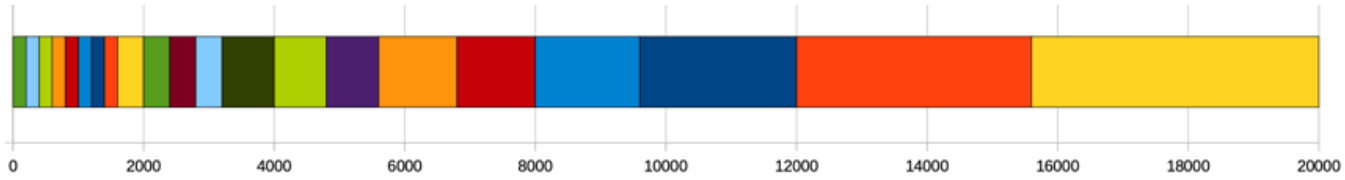


图4 CELT中Bark谱的频带分布

从 DCT 系数中去除未量化频段的能量值，产生的剩余信号的系数部分是通过金字塔矢量量化（PVQ）来编码的。这种编码方式使用固定（可预测）长度的码字，从而对比特位错误

有更好的鲁棒性。CELT 将频带的能量分为粗粒度能量和细粒度能量，并且两者相加为整体的频带能量。而在编码时可根据想要编的码率来调节粗细力度的码率分配，这样就可以兼容不同的码率设置。具体 CELT 的编码流程如图 5 所示：

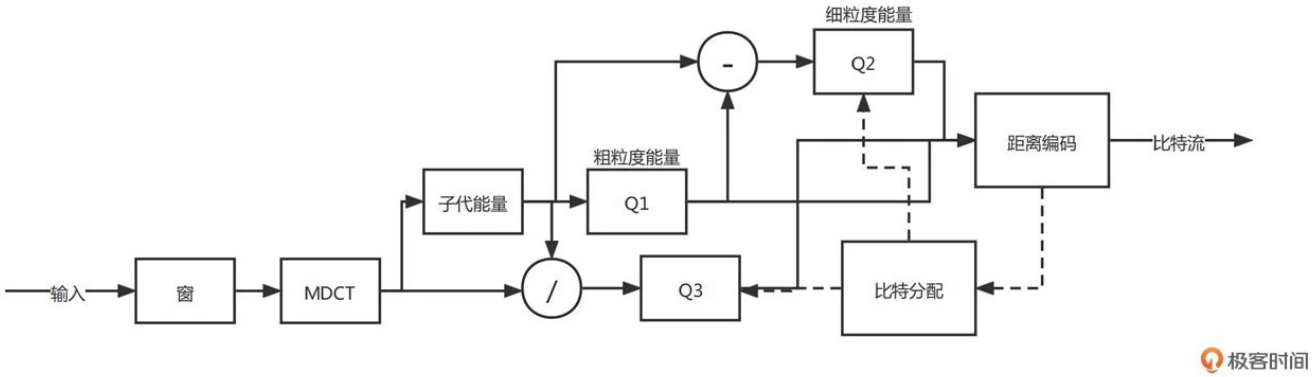


图5 CELT的编码流程

小结

这一讲我们主要介绍了一些音频编 / 解码的基本原理。音频的编 / 解码主要有语音编 / 解码、音乐编 / 解码和混合编 / 解码。其中通用的编 / 解码方法有非线性量化、减少采样率、矢量量化等方法。对于语音信号可以使用声道模型建模，然后采用线性预测加残差的方法来进行编 / 解码，而对音乐信号则需要根据人耳敏感频响曲线分频带来建模。

其中，在 WebRTC 中默认使用的编 / 解码器 OPUS 就是这类编 / 解码器。其语音主要采用 SILK 来编码，音乐则采用 CELT 来编码。在 OPUS 内部还有一个基于人工神经网络的音乐判断器来进行人声和音乐的自动切换，从而达到最好的编 / 解码效果。

其实在音频编 / 解码领域有很多成熟而优秀的编 / 解码器，初学者不要被这些复杂的数学、声学、心理学知识所吓到。我们完全可以在掌握编 / 解码器原理的基本概念之后根据自己的场景，选择一些成熟的编 / 解码器来使用，具体怎么选我们将会在下一讲中详细介绍。

思考题

如果我们人类每分钟最多说 200 个字，假设一共有 4096 个常用汉字。设想一下如果采用文字识别作为编码（语音转文字 ASR），然后用文字转语音技术（TTS）来解码那这种编 / 解码器的码率会是多少？提示一下，4096 可以用多少 bit 来表达？

欢迎留言和我分享你的答案和疑惑。我们下节课再见！

分享给需要的人，Ta订阅后你可得 20 元现金奖励

生成海报并分享

赞 0 提建议

© 版权归极客邦科技所有，未经许可不得传播售卖。 页面已增加防盗追踪，如有侵权极客邦将依法追究其法律责任。

上一篇 08 | 回声消除算法实践指南

精选留言 (2)

写留言

坚坚

2021-12-10

2的12次方，可以用12bit储存，即1.5BYTE,1.5*200=300Byte每秒，若进一步考虑句子相关性，例如通过词来编码，或者进一步利用上下文信息来编码，若采用词向量的方式，应该还可以进一步压缩，这个应该就是属于用深度学习来做编解码的范畴了。

展开

springXu

2021-12-10

4096种汉字共需要4k的编码。2的12次方。 一分钟200个字，也要800k的码率。。。 这个和文中提到的48000的采样率16bit的编码768k的码率还要大。

展开